



1 SUMMARY

Given a character string and a search character **locates the first occurrence of the character in the string** or optionally locates the **first non-occurrence** of the character.

In either case the search may be in a forward direction from the beginning or in a backward direction from the end of the string.

ATTRIBUTES — **Version:** 1.0.0. **Types:** IC01A. **Original date:** October 1971. **Origin:** M.J.Hopper, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Argument list and calling sequence

To search for the occurrence of a character

```
IPOS = IC01A(STRING,LENGTH,CHAR)
```

To search for the non-occurrence of a character

```
IPOS = IC01B(STRING,LENGTH,CHAR)
```

STRING is a CHARACTER variable (or array) which the user must set to the string to be searched, see §2.2.

LENGTH is an INTEGER variable which is used to tell the subroutine the direction of search and the length of the string. Let $n \geq 0$ be the length of the string. Then if the search is to be from the beginning of the string in a forward direction the user must set $LENGTH = n$. If the search is to be made backwards from the end of the string the user must set $LENGTH = -n$. If $LENGTH = 0$ the subroutine returns a value of zero.

CHAR is a CHARACTER variable. When using IC01A the user must set CHAR to the single character to be located. When using IC01B it must be set to the character to be ignored in the search, see §2.2.

IC01A is an INTEGER function subprogram and its value will be set to the position of the character relative to the beginning of the search string, or zero if the subroutine fails to find the character.

IC01B is an INTEGER function subprogram and its value will be set to the position of the first non-occurrence of the character relative to the beginning of the search string, or zero if the subroutine fails to find a character different from the search character.

2.2 The argument types

The subroutine takes the first byte of CHAR and searches the first LENGTH bytes of STRING.

3 GENERAL INFORMATION

Use of common: None.

Workspace: None.

Other routines called directly: None.

Input/output: None.

5 EXAMPLE OF USE

Consider the problem of retrieving a text field enclosed in parentheses from an input line, and suppose that it is also necessary to strip off leading and trailing blanks from the text, e.g. given the input statement

```
( HSL LIBRARY )
```

extract the text 'HSL LIBRARY'. The following code would do

```

CHARACTER*80 C           to hold the input line
CHARACTER*20 NAME       to hold the name
C
  READ(5,10) C           read in input line
10 FORMAT(A)
C
  FIND THE OPEN PARENTHESIS
  IP1=IC01A(C,80,'(')    will find it in 2nd column
                        error if '(' not there
  IF(IP1.EQ.0) GO TO 90
C
  FIND THE CLOSED PARENTHESIS
  IP2=IC01A(C,80,')')    will find it in 17th column
                        error if ')' not there
                        error if in wrong order
  IF(IP2.LE.IP1) GO TO 90
C
  FIND FIRST NON-BLANK FROM '('
  IC1=IC01B(C(IP1+1),80-IP1,')')+IP1    should be column 4
C
  FIND FIRST NON-BLANK BACK FROM ')'
  IC2=IC01B(C,-(IP2-1),')')            should be in column 14
  IF(IC2.LT.IC1) GO TO 90              error if parentheses is empty
C
  COPY STRING INTO NAME
  NAME=C(IC1:IC2)                      copy string into NAME
  - -
C
  ERROR HANDLING
90 - -

```